

Perguntas frequentes

Para implementar a redirecção de saídas pedida pelo ponto 1.2 do enunciado, é correto usar `fclose(stdout)` antes de substituir a entrada respetiva na tabela de ficheiros abertos por uma outra?

Não deve ser usado `fclose(stdout)`. Devem considerar as alternativas usando `close/dup` ou `dup2`. As respetivas páginas do manual providenciam mais informação.

Deve ser impressa uma linha no `log.txt` a descrever os comandos que não foram executados com sucesso (e.g. debitar de conta sem saldo suficiente)?

Ambas as alternativas são aceitáveis (ou seja, nenhuma será penalizada): ou imprimir apenas para os comandos executados com sucesso, ou imprimir para todos os comandos (independentemente da execução ter ou não tido sucesso. Em qualquer caso, a impressão só deve acontecer depois de executado o comando.

No caso de comandos sobre contas distintas, a ordem de impressão no `log.txt` deve reflectir a ordem de tempo real com que esses comandos terminaram?

Não. Apenas é necessário assegurar esse requisito para comandos que trabalhem sobre a mesma conta. Idealmente, a solução deverá evitar introduzir mecanismos de sincronização adicionais (isto é, tentar reutilizar os mecanismos de sincronização já implementados até ao momento).

A escrita no `log.txt` por parte de cada tarefa trabalhadora deve ser atómica (ou seja, a linha impressa não deve ser interrompida por linhas impressas por outras trabalhadoras concorrentes)?

Embora seja um requisito avançado, sim, é desejável que assim seja. Havendo várias soluções possíveis, a mais simples será abrir o ficheiro com a função `open` e escrever com a função `write` (a função `write` garante escritas atómicas na grande maioria dos sistemas UNIX/Linux).

Sugestão: compor uma *string* usando `snprintf`; depois passar essa *string* como argumento à função `write`.

Como deve ser obtido e impresso o "TID" a imprimir no `log.txt`?

Uma alternativa (entre outras) é usar a função `pthread_self` e converter o retorno (do tipo `pthread_t`) para uma *string* usando a função `snprintf` com o formato `%lu`. Embora essa solução não seja recomendada pela especificação POSIX, será aceite como correta na avaliação do projeto.

Devem as operações sobre contas executadas pelo processo de simulação também constar no `log.txt` mantido pelo i-banco?

Como é dito no enunciado, apenas as operações executadas pelas tarefas trabalhadoras deverão ser impressas no log.txt. Isto exige que o processo filho que chame as funções debitar/creditar/lersaldo durante a simulação não tente escrever no log.txt. Uma das várias alternativas possíveis para restringir essa escrita é simplesmente: Dentro do programa executado pelo processo filho, atribuir à variável/argumento que contém o *file descriptor* que identifica o ficheiro log.txt (que foi aberto pelo pai) um valor inválido (por exemplo, -1). Desta forma, o código das funções debitar/creditar/lersaldo pode conferir o valor desse *file descriptor* para decidir se efetua a escrita (caso de *file descriptor* válido) ou não (caso de *file descriptor* inválido). Existem outras soluções igualmente corretas e que serão, naturalmente, aceites sem penalização.